



Stankaitis P, Iliasov A, Romanovsky A. (2017)

[A Railway Simulation Suite For Modelling Advanced Railway Control Systems.](#)

International Journal of Transport Development and Integration, 1(3), pp. 558-567.

Copyright:

© 2017 WIT Press, www.witpress.com

DOI: 10.2495/TDI-V1-N3-558-567

URL: <https://www.witpress.com/elibrary/tdi-volumes/1/3/1431>

Date deposited:

04/01/2017

A RAILWAY SIMULATION SUITE FOR MODELLING ADVANCED RAILWAY CONTROL SYSTEMS

P. STANKAITIS, A. ILIASOV & A. ROMANOVSKY
Centre for Software Reliability, Newcastle University, UK.

ABSTRACT

All large railway networks use a mixture of outdated, modern and emerging signalling and train operation principles. There is a need to develop novel modelling and verification mechanisms to support mixed traffic scenarios, including, for example, mixing different types of signalling and driving. In our previous work we introduced the Unified Train Driving Policy (UTDP) formal modelling language for uniformly capturing diverse signalling principles and mixing, in a demonstratively safe manner, at the node and/or network-level novel and legacy signalling principles. This paper describes our work on making UTDP practical and useful for the engineers.

Keywords: advanced railway control systems, simulation, formal methods, automatic train operation (ATO), Unified Train Driving Policy.

1 INTRODUCTION

1.1 Background and literature review

The transport sector has been seriously affected by the economical progress in many countries (Abril et al. [1]). The traffic congestion and high negative environmental impact in the metropolitan cities are the key reasons behind the global initiative to improve railway and make it more competitive. A number of research projects (e.g. [2, 3]) were launched to investigate the current problems of the railway industry and look into possible solutions.

One of the EU-funded railway projects [2] identifies the effective management and optimization of railway networks as a grand challenge for the future of railway. There is a number of issues making it difficult to address. First of all, there is a correlation between capacity and service reliability. For example, for the heavily congested metropolitan city's rail networks, which are run close to the maximum capacity, ensuring service reliability becomes critical. Paper [1] shows a clear correlation between capacity and service reliability, where adding extra trains exponentially increases average delays in the network. Furthermore, ensuring the efficient use of energy in the railway network is crucial too. For example, González-Gil et al. [4] show this for maintaining urban rail as a solution for transport problems.

Significant network improvements can be achieved by upgrading the current railway infrastructure. The European Rail Traffic Management System (ERTMS) is a leading initiative to overcome critical railway problems. A modern train control system and continuous radio-based communication are the two key components of the new management systems. Optimization of capacity will be achieved by using systems capabilities to reduce the block lengths, or even by switching in yet to be implemented a moving block signalling.



This paper is part of the proceedings of the 15th International Conference on Railway Engineering Design and Operation (COMPRAIL)
www.witconferences.com

Further improvements in capacity, energy and stability can be achieved by fitting trains with the Automatic Train Operation (ATO) system, which allows optimization of train speed profiles. Even though these modifications would dramatically optimize railway network use of resources the deployment of adjustments can take a long time and be very costly.

In the short term, a more cost-effective solution is to increase the capacity of the existing railway by utilizing computer-based simulation techniques [1]. The railway companies use proprietary, or externally developed, simulation suites like OpenTrack [5] and RailSyS [6] in a timetable and infrastructure optimization. A visual and user-friendly aspects of simulation platforms offer an intuitive method to analyse railway network without requiring a tedious simulator configuration process.

As discussed by Stankaitis et al. [7] any type of improvement of railway systems has to meet the paramount requirement of safety. Railway certification standards require applying semi-formal or formal methods for description of the system and even correctness validation. The industry has been successfully applying formal techniques for reasoning about safety for some time now with many developed systems currently in operation. Perhaps the best example was the use of the B-method Abrial [8] in major railway projects including a Paris metro line and a more complex New York Canarsie line (Essame et al. [9]). However, simulation techniques are still extensively used by the railway industry to ensure that safety standards are preserved within system. And even though simulation tools are progressively becoming more powerful the main drawback of this technique is a small state-space coverage. In contrary to the simulation approach formal safety verification of railway networks is a complicated process, which requires expertise in the mathematical modelling; however, it provides techniques to guarantee safety for a complete system state-space.

A significant effort has been put by academic research projects like SafeCap [10] into hiding mathematical complexity to make it easier for railway engineers to use formal methods. The SafeCap project was started with an objective to develop techniques for overcoming railway capacity problems without weakening safety constraints. The key outcome of the first stage of the project was a SafeCap tooling environment, which allowed railway engineers not only to design and analyse railway junctions/stations, but also to formally and automatically verify their safety.

A major challenge of optimizing the use of railway network resources like capacity and energy without undermining safety constraints or minimizing service reliability for large-scale industrial projects requires state-of-the-art techniques and tools. The user-friendly railway simulation platforms currently used by the industry are typically rigid. Advanced functionality and expressiveness of simulation suites are sacrificed for its simplicity. Railway academic community has successfully analysed and applied a variety of analytical, iterative methods and even genetic algorithms to optimize railway networks. However, our analysis of a number of academic research projects shows a lack of adequate and multi-purpose simulation tooling for the railway domain. A large number of projects (e.g. [11, 12]) have used popular and powerful suites like OpenTrack or RealSyS to design or evaluate novel algorithms or optimization methods. Yet in many instances researchers had to conduct a manual experiment reconfiguration of the tools to test the newly developed methods. Some other projects developed single-purpose railway simulators for their experiments.

There has been few attempts to create a more expressive and flexible simulators. For example, Grube et al. [13] developed a dynamic event-driven railway simulator for metro systems. The developed railway simulator enabled users adjusting a number of parameters in order to model complex metro systems. Furthermore, using a widely applied software

package, Matlab users were able to experiment and design novel control and optimization algorithms. Paolucci and Pesenti [14] attempted to exploit object-oriented paradigm and design a discrete-event simulator for underground railway systems. The ability to define a network of relationships between physical and decisional objects allowed users to design a whole system. The base of a code and simulation code were automatically generated by the simulation suite. However, the research group was yet to make the tool more usable by implementing a functionality to modify system objects and processes.

The application programming interface (API) enables some tools, like Open-Track, to be connected to other third-party tools to add extensions, implementing, analysing or developing train control systems or dispatch algorithms. But configuring the software may sometimes be tricky, or tools can be simply incompatible. In the next section we present the SafeCap project approach to allowing optimization of railway networks by developing an expressive simulator.

1.2 Modelling advanced railway systems

The SafeCap team at Newcastle University has been working on railway modelling and formal verification for nearly six years. One of the topics we have been looking at recently is the definition, simulation and validation of novel, hybrid or simply exotic signalling principles. In our previous work Iliasov et al. [15] introduced the Unified Train Driving Policy (UTDP) (Fig. 1), a generic approach that can capture both existing and novel signalling as well as scenarios involving a mixture of signalling solutions and transient arrangements like virtual trains and platoons. The primary subject of this formal language is the definition of laws of train movement that result in a safe railway operation, achieving optimum capacity. The reasoning about capacity at the same level as safety means that one is able to formulate a range of QoS criteria such as average train speed, minimum speed at a given location, maximum separation between trains and so on.

In UTDP, all stateful parts of railway are known as actors. A train and a point (a track switching device) are two possible examples of an actor. New actor types may be defined in a model (i.e. a station controller) and all actors are permitted to interact with each other

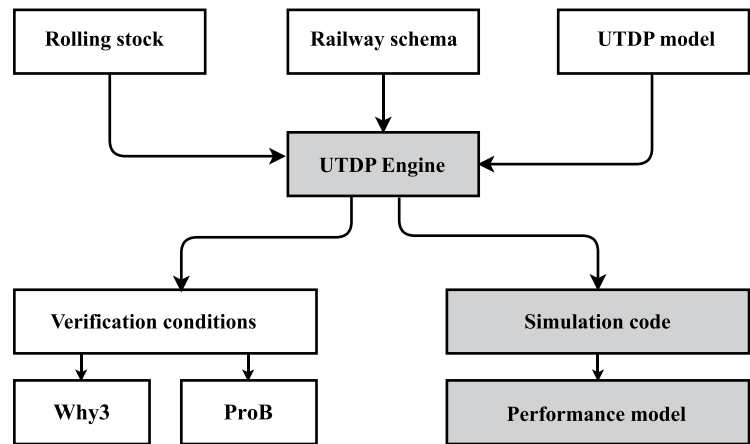


Figure 1: The UTDP approach with highlighted executational side.

through some shared state. An actor lifecycle is split into three stages: continuous evolution (e.g. a train moving in accordance with the laws of physics), mutation (a step-change due to a logical trigger) and hibernation (inability to evolve or mutate). An evolution of a train is described by a continuous or piece-wise continuous function of the effective longitudinal force. It defines a smooth progression from one position to another and accounts for such effects as air drag, gradient, rolling resistance, variable tractive effort and others. A mutation is a change of law governing an actor evolution; it may be dictated, for instance, by signalling logic or speed regime. An actor goes into hibernation when it is unable to evolve (a train waiting for a proceed signal).

At the core of UTDP is a formal notation used to express both static and dynamic properties of railway. The static part is based on a formal domain-specific language and captures railway topology, route boundaries and track-side equipment. The dynamic part describes train properties and the way trains move over the track.

In Fig. 2, a railway schema example and an excerpt of its signalling in UTDP are given. The layout is a simple junction with a fixed block signalling. In fixed block signalling, a simple programme (called a control table) helps a driver to maintain safe speed. The programme observes the infrastructure state (points and train detection circuits) and interacts with a driver via track-side signals that communicate commands to a driver via colour-coded messages. If a driver obeys the commands then, assuming the control table is correct, it may be shown that a train is never within a dangerous proximity to another train. UTDP is not specific to fixed block signalling; one may express various forms of the moving block principle.

In this paper, we focus on executional semantics of UTDP shown in Fig. 1, which is exactly the simulation of railway operation. In our research, we propose to develop a new simulator on making UTDP practical and useful for the engineers.

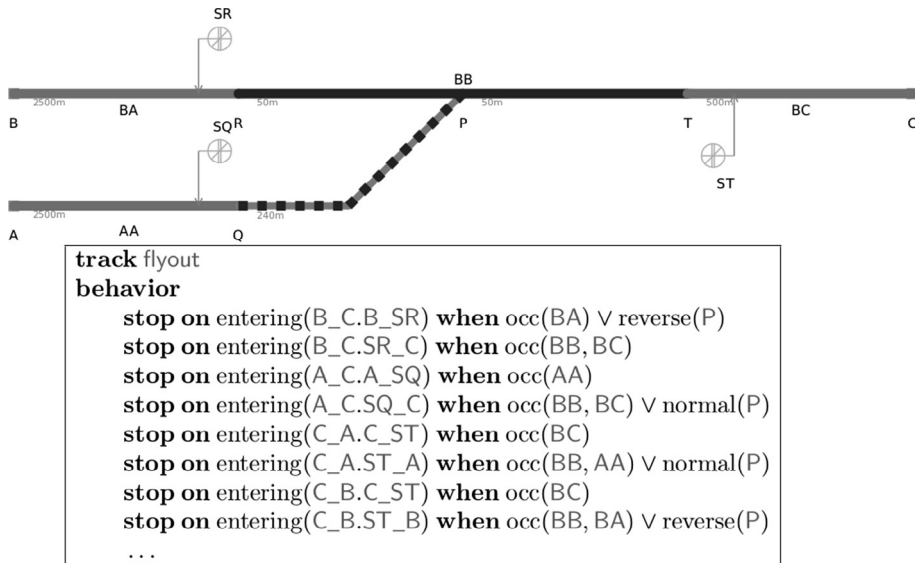


Figure 2: In the layout, the valid paths through the junction are A_C, B_C, C_A and C_B; signals SR, SQ and ST define routes B_SR, SR_C, ...; AA, BB, ... are the train detection circuits that report the presence or absence of a train; point P is located on train detection circuit BB; finally, R, Q and T are the labels of train detection circuits boundaries.

2 SIMULATOR PROGRAMMING AND IMPLEMENTATION

The traditional approach to railway simulation consists in time or distance slicing where within small time or distance extents train movement laws are fairly straightforward and determined only by the boundary conditions. A more advanced version could use an adaptive step size to maintain a degree of modelling fidelity.

Still such technique has a number of drawbacks. One must take care that errors do not accumulate over time (since in machine arithmetic $3 * x$ is not necessarily the same as $x + x + x$, especially for small values of x) and important but short-lived phenomena are accounted for (e.g. a delay in train occupation confirmation). According to the Shannons theorem, the time delta of a time-slicing simulator must be at least two times smaller than the extent of the shortest event and this means time delta of around 0.01 second.

The original SafeCap simulator [16] took a different approach of event-based simulation where the temporal extent of a simulation step is determined by the amount of interference between various entities of a simulated railway: trains, signals, points and virtual entities such as route reservation logic. Such a technique does not have fidelity limitations and has a far greater potential performance than a naive time-slicing simulator. It also is far better compartmentalized in actor-like entities that dictate operational rules. For instance, the determination of the most restrictive target of train is done by collating and processing a list of actors affecting train speed. Actors may be added and removed, giving a compositional approach to train behaviour modelling.

Also the original simulator is flexible and extensible; the changes still have to be implemented at the level of source code of the simulator. To test some speculative scenarios it was deemed necessary to have an even more general approach to simulation without disrupting the code base of the existing simulator.

Traditionally, driver behaviour is simulated by driving a train to some percentage of its overall acceleration with braking capability that is always within the safety conditions of signalling and interlocking. For out-of-the-box scenarios, signalling rules stand in the way; so to enable the freedom of experimentation we decided to scale back the new simulator to the very bare bones and to use a more basic, though more expressive, time-slicing approach.

Let us consider what a train is. A train driver has but one essential form of control over train: acceleration and braking (assuming both are never together). From a high-level perspective, it is a function of many variables including the capability, position and speed of the controlled and nearby trains, infrastructure state, weather, staff rotation and so on. For any such set of variables this function yields just one number: the effective acceleration (a negative value representing braking). Somewhat simplistically we could try to summarize the statement as a set of simultaneous equations defining train acceleration control TC and infrastructure control IC (the time domain t is implied, \mathbf{T} is a set of all train states, \mathbf{C} is infrastructure state, a subscript is a train name and superscript is a discrete time step):

$$\begin{cases} a_i = TC_i(T, C^j, \dots), \\ c^{j+1} = IC(T, C^j, \dots) \end{cases} \quad (1)$$

The starting point of every simulation scenario is then just a layout made of tracks, points and, optionally, detection circuits, with some trains appearing on the boundary nodes according to some pattern. Such a starting point is clearly unsafe: there is nothing preventing train collision or other safety violations. However, the driver behaviour may be specified to an unlimited level of detail and a variety of actors (trains, points, signals or some new kind

of actors) are free to share information and exchange messages. The goal is to programme a driver who would, at least for a variety of simulation scenarios, move a train safely across a layout. The driver's logic is defined simply as a function of the form

```
double acceleration(World world) {
    ...
}
```

The function computes acceleration or braking in the context of the current world, a shared data structure describing both current and other trains position, acceleration as well as the state of other actors, i.e. points. A simple yet unsafe driving function is

```
double acceleration(World world) {
    return 0.5;
}
```

It prescribes a constant acceleration of 0.5 m/s^2 and disregards any obstacles or hazards. A more realistic one would have to assess the situation on the train horizon, in particular state of the points, trains occupying same paths or appearing from converging paths, as well as to comply to scheduled stops and fixed speed limits. The conceptual difference is that a safety envelope is taken away. The logical entity of a driver is responsible both moving a train forward (liveness) and ensuring absence of collisions and derailments (safety).

A train is but one kind of actor. There are predefined actors for a signal, point and train detection circuit. Generic actors may be defined to cater for and structure new kinds of signalling logic. One example is an actor commanding route reservation and point locking on the basis of a timetable. For a system to function as a whole, all the constituent actors need to coordinate with each other. This is achieved either through accessing shared state directly or via message passing.

3 METHODOLOGY EVALUATION

In this section, we present an application example we used to assess a time-slicing simulator. The primary objective was to design a novel train control mechanism and evaluate its performance with respect to the original simulator with a route-based signalling using a simple bridge layout shown in Fig. 3.

3.1 Experiment description

To demonstrate the control system's adaptability to various service patterns as part of the new simulator suite we designed a scenario generator. The tool in a non-deterministic order and frequency would construct a train actor, which would appear on boundaries of the layout. The generated train actor would randomly inherit several parameters including a train class, which describes train physical properties and a route.

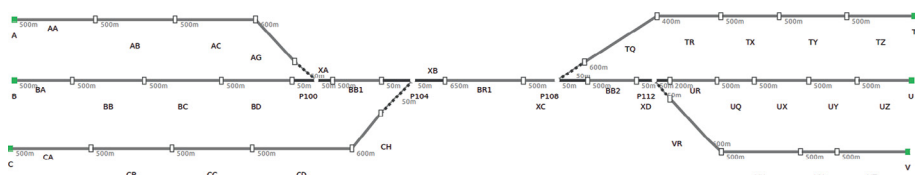


Figure 3: A bridge layout used to evaluate train control algorithm.

In order to demonstrate the flexibility of the designed control system several scenarios were simulated. We evaluated the application example by performing identical train service patterns in the original simulator, which used a route-based signalling, and in the newly developed train control system. The performance was compared using the capacity measurement together with the analysis of train speed profiles.

In the following sections we provide a more detailed description of the control algorithm, capacity results and tool evaluation.

3.2 Train control algorithm

To design a more dynamic and optimal train control mechanism one cannot rely on physical signals any longer. Such signalling system would require a precise and continuous knowledge of the railway network status including the rolling stock and infrastructure parameters. In our design a control system has a full autonomy to manage both trains and the infrastructure. A similar way of thinking is used in ERTMS Level 3, where a centralized unit called Radio-Block Centre receives a continuous information about the network and issues a movement authority.

To understand our algorithm let us reconsider eqn (1). The first technical obstacle is that any meaningful approximation of TC yields non-linear differential equations while infrastructure control by its nature is decidedly discrete. The historical perspective on the problem is to replace the computation of train acceleration with a simpler task of the determination of a safe upper boundary

$$a_i \leq SA_i(T, C^j, \dots). \quad (2)$$

This style of thinking is used, for instance, in ERTMS Level 3 to maintain train acceleration within some sensible braking curve. Computing such SA while maintaining confidence in its safety properties is quite challenging and a meaningful discretization requires a high fidelity proximity or position sensing. When such means are unavailable one has to rely on a cruder technique where the control variables are discretized in temporal or spacial domain. Direct acceleration control then becomes too finely grained a tool for the task and bounds are normally given in terms of train velocity

$$v_i^{k+1} \leq SV_i(T^k, C^j, \dots). \quad (3)$$

The safe velocity bound used in our control system was based on a moving block signalling principle. A control system would continuously compute speed, V_i , to maintain a safe distance between trains and avoid train derailment over unset point or unsafe curve speed. Further system's performance optimization can be achieved by integrating a dynamic speed adjustment function. A typical example is over-speeding trains, which usually results in situation, where train is required to stop and wait before crossing the point. As our system is able to access any information about the network, train speed can be adjusted with respect to other trains more dynamically.

The second essential part of the control algorithm is the train scheduling sub-system. Scheduling algorithm is mainly responsible for solving the use of shared resources (e.g. points) issue. A variety of aspects can be considered in the scheduling function. For example, a simple yet widely used approach in queuing algorithms in various domains is a First in, First out method.

The bridge is a topologically limited layout with a heavily used single line. In order to obtain a better performance one would like to schedule trains, so the fastest trains would have a priority over sections XB, BR1 and XC in Figure 3. So, we considered a physical distance aspect and used it to improve the system. Since we constructed an experiment with randomly appearing trains on the boundaries no pre-computation of optimal schedule is done. Nonetheless, the scheduling system attempts to optimize the performance, though; if we were to plot scheduling algorithm output, one would notice oscillations for a short period before steady-state is reached.

3.3 Capacity and speed assessment

To evaluate the design of the developed train control system we compared its capacity performance against the fixed block signalling system. The definition of railway network capacity is a highly debatable subject. In our experiment, a *used capacity* metric described in [1] was applied. Often, a *used capacity* measurement results in a slightly lower value from the practical capacity; however, it does not require a rigorous computation.

In order to clearly show our systems advantage we needed to demonstrate performance improvements for a large number of scenarios. Therefore, we the tool generated multiple service patterns and applied them to both systems. The experiment results are shown in Fig. 4, where in an individual experiment *used capacity* number is plotted against the capacity for both signalling systems. The performance of the novel train control system is represented by red points, whereas the fixed block system by blue.

Results clearly demonstrate that a developed control system, which was based on a moving block signalling principle, outperforms a traditional fixed block system. Noticeably, there are some improvement deviations; for example experiment 2 has a relatively lower improvement coefficient. Throughout the study we noticed that greatest improvements are seen when a large number of trains run on a frequent service.

Furthermore, we wanted to investigate the effect of dynamic speed adjustment functionality we integrated in the speed calculation function. Previously, we discussed an over-speeding train scenario, where a train is required to stop and allow another train to pass. This was captured in one of the simulation runs with a fixed block route-based signalling system; the speed profile is shown in Fig. 5 (red plot). The identical scenario simulated in the system with the newly developed control system produced a blue speed profile. The latter train was

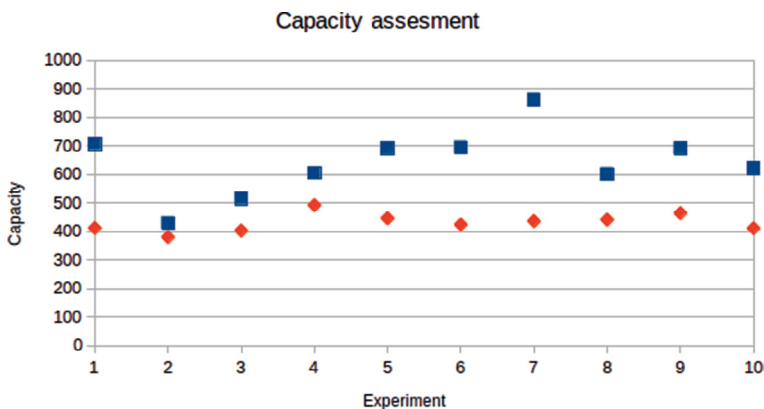


Figure 4: A capacity evaluation of two train control systems.

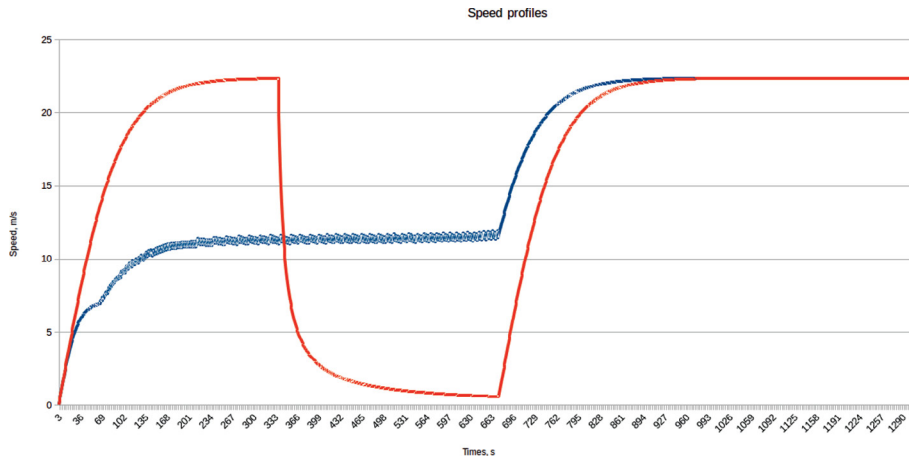


Figure 5: A speed profile comparison of control systems.

running at lower velocity before crossing the point and was able to enter a new section from a running state, and thus, reduce a route completion time.

One of the ongoing work packages in the SafeCap project investigates the energy aspect of the railway. In the future, we plan to integrate energy consumption plug-in, which could analyse speed profiles like the one shown in Fig. 5, and provide essential information for railway engineer.

All these measurements were produced using a new simulator, which is now integrated into the SafeCap platform. The platform, also has a replay functionality, which can be used to further investigate the model. A further use analysis of the simulator is given in the following section.

3.4 Simulation tool evaluation

A lack of adequate railway simulation tools restrains our progress for improving development of railway networks. In our research, we designed a vigorous simulation tool in attempt to make UTDP approach for modelling complex railway networks practical.

We demonstrated capabilities of the new simulation suite by designing a novel train control system. However, to fairly assess the simulation platform the effort factor required to develop a new control system has to be considered. To simplify the development phase, tool provides a number of functions for computing a variety of railway network parameters, which can be easily called in the development framework. Moreover, a user-friendly graphical interface can provide user with important information about the capacity or speed profiles, as well as can visually replay a scenario. Although, the simplicity aspect of simulation tool utilized in many commercial tools like OpenTrack and RailSys is reduced, as, the present use of the simulator requires some basic programming knowledge.

4 CONCLUSIONS

An expressive time-slicing simulator was developed in an effort to make the formal approach [15] for modelling novel signalling systems feasible. A simple application example of the simulation suite was demonstrated by developing an exotic train control system, performance of which was compared to the fixed block signalling system. As authors pointed out, the simulation platform is suitable for a variety of railway development and optimization purposes.

However, the current implementation of the simulation tool requires a basic programming knowledge. In order not to undermine expressiveness of the tool, yet make it more usable for railway engineers, two future work directions were identified. A signalling library with existing and novel train control mechanism will be developed to allow engineers to design more complex railway networks with heterogeneous signalling. The second direction is to link UTDP modelling language and object-oriented programming language currently used by the simulation platform. Finally, in the future we intend to focus on safety verification aspect of the approach. However, in order to be able to verify advanced train control systems formal semantics have to be revisited.

ACKNOWLEDGEMENTS

This work is supported by the RSSB SafeCap+ project. We are grateful to our colleagues from Siemens Rail Automation for invaluable feedback. We also thank Roberto Palacin and Luke Martin for their suggestions and comments.

REFERENCES

- [1] Abril, M., Barber, F., Ingolotti, L., Salido, M.A., Tormos, P. Lova, A., An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44, pp. 774–806, 2008.
- [2] IMPROVERAIL project. *Deliverable D10 of the project*. 2003.
- [3] European Commission. *EU Policy and Its Impact on the Rail System*. 2006.
- [4] A. González-Gil, R. Palacin, P. Batty, J.P. Powell., A systems approach to reduce urban rail energy consumption. *Energy Conversion and Management*, 80, pp. 509–524. 2014.
- [5] OpenTrack Simulation Platform, www.opentrack.ch
- [6] RailSyS Simulation Platform, www.rmcon.de
- [7] Stankaitis, P., Iliasov, A., Safety Verification of Heterogeneous Railway Networks. *International Conference: Reliability, Safety and Security of Railway Systems*. 2016.
- [8] Abrial, J.R., *The B-Book*. Cambridge University Press. New York, NY, 1996.
- [9] Essame, D. Dolle, D., B in Large-Scale Projects: The Canarsie Line CBTC Experience. *LNCS*, vol. 4355. Springer, pp. 252–254, 2007.
- [10] Iliasov, A., Romanovsky, A., SafeCap domain language for reasoning about safety and capacity. *Workshop on Dependable Transportation Systems at the Pacific-Rim Dependable Computing Conference*. 2012.
- [11] Luethi, M., Nash, A., Weidmann, U., Laube, F., Wuest, R., Increasing Railway Capacity and Reliability through Integrated Real-Time Rescheduling. *Institute for Transport Planning and Systems*, ETH Zurich. 2007.
- [12] Li, Wei., Tang, Tao., The impact of signalling system heterogeneity on urban transit capacity. *IEEE International Conference on Intelligent Rail Transportation (ICIRT)*, pp. 123–127, 2013.
- [13] Grube, P., Núñez, F., Cipriano, A., An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network. *Simulation Modelling Practice and Theory*, 19, pp. 393–405, 2011.
- [14] Paolucci, M., Pesenti, R., An Object-Oriented Approach to Discrete-Event Simulation Applied to Underground Railway Systems. *Transactions of the Society for Modelling and Simulation*, 72(6) pp. 372–383, 1999.
- [15] Iliasov, A., Lopatkin, I., Romanovsky, A., Unified Train Driving Policy. In *Formal Methods Applied to Complex Systems*, eds. J.-L. Boulanger, Wiley, pp. 447–473, 2014.
- [16] SafeCap Simulation Platform, www.safecap.co.uk